

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой
О.В.Непомнящий
_____ инициалы, фамилия
подпись
« ____ » _____ 2018 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника

код и наименование направления

Разработка веб-сервиса встраиваемых тестов

тема

Руководитель

подпись, дата

доцент
канд.тех.наук.
должность, ученая
степень

А.В.Редькин
инициалы, фамилия

Выпускник

подпись, дата

А.А.Шатоба
инициалы, фамилия

Нормоконтролер

подпись, дата

доцент,
канд.тех.наук.
должность, ученая
степень

В.И. Иванов
инициалы, фамилия

Красноярск 2018

Содержание

Введение.....	3
1 Анализ технического задания.....	4
1.1 Анализ существующих аналогов.....	4
1.2 Разработка системы.....	11
2 Проектирование.....	12
2.1 Модули системы.....	12
2.1.1 Интерфейс.....	13
2.1.2 Сервер.....	15
3 Реализация.....	16
3.1 Используемые инструменты.....	16
3.1.1 Инструменты интерфейса.....	16
3.1.2 Инструменты сервера.....	18
3.2 Описание работы системы.....	19
3.2.1 Интерфейс.....	19
3.2.2 Сервер.....	22
3.3 Описание базы данных.....	24
4 Результат работы.....	27
Заключение.....	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	31

ВВЕДЕНИЕ

В настоящее время количество веб сервис непомерно растет и нередко возникает потребность провести опрос или тест. Это необходимо для того, чтобы определить, какая аудитории посещает данный ресурс или для того, чтобы узнать, что думает аудитория о прошедшем событии или мероприятии.

Одна из типичных задач, возникающих при разработке веб-сервисов это тестирование, опрос аудитории. Нередко такая задача возникает при создании простой веб-странички. Что же делать в таком случае? В данной ситуации возможны два варианта решения проблемы. Либо реализовать такую систему самому, что занимает большое количество времени и ресурсов. Но гораздо удобнее и менее затратно, воспользоваться разрабатываемой системой или уже готовыми системами.

Разрабатываемая в данной работе система призвана решить эту проблему. Для ее решения необходимо посетить данный сервис, создать требуемый опрос или тест, и при помощи полученной ссылки встроить в свой веб-ресурс.

В настоящей работе решается задача создания сервиса с возможностью встраивания в другие ресурсы модуля теста или опроса, с несложным, интуитивно понятным интерфейсом. Необходимо создать сервер, который способен обрабатывать поступающие запросы, обращаться к базе данных, обрабатывать результат, полученный от базы, и отправлять данные интерфейсу. Сервер способен работать с данными, которые пришли от интерфейса, обрабатывать их, сохранять результат в базу, и сообщать интерфейсу об этом.

Целью данной работы является разработка веб-сервиса, который поможет создавать собственные опросы или тесты, избавит пользователя от создания собственного сервиса тестирования.

1 Анализ технического задания

1.1 Анализ существующих аналогов

Предлагаемая к использованию система не является уникальной, существуют несколько ее аналогов, одним из таких является сервис SurveyMonkey. SurveyMonkey- это система, в которой совершенно любой пользователь способен создать свой тест, опрос по какому-то продукту или по проведенному мероприятию, также любой пользователь способен проанализировать данные других опросов и тестов [1]. Данная система имеет привлекательный интерфейс, большой функционал и возможности.

При создании теста у пользователя есть выбор, каким образом он хочет получать статистику по данному тесту: на почту или в социальную сеть. В данной системе реализовано ведение статистики, по созданному вами тесту имеется несколько критериев это: сколько человек прошло ваш тест, какая у вашего теста популярность (рисунок 1).

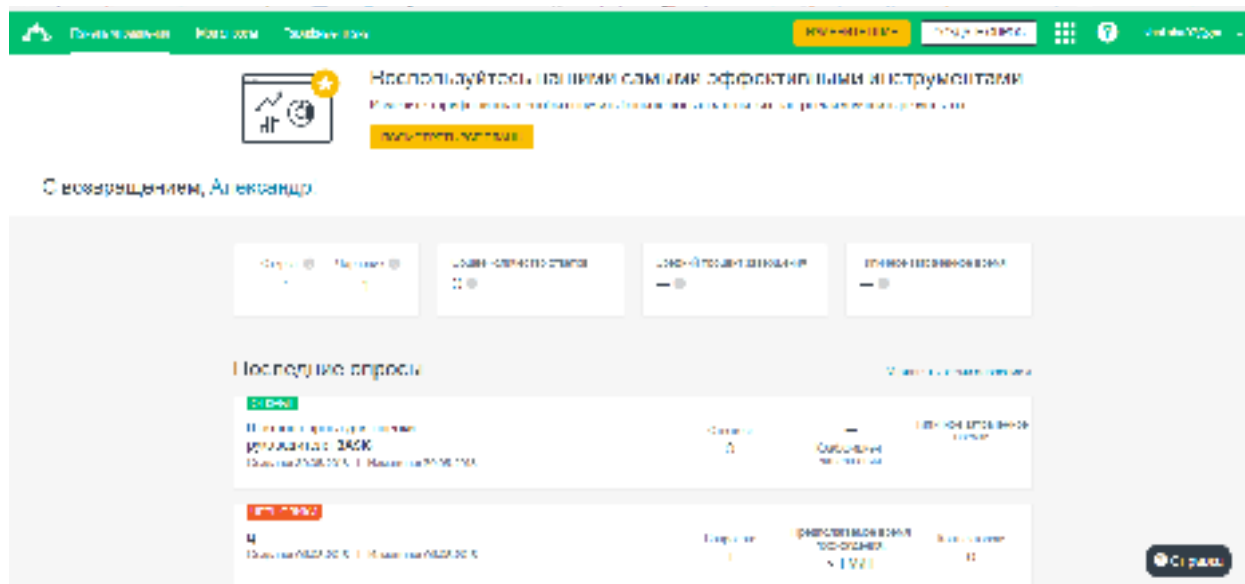


Рисунок 1 – Профиль пользователя со статистикой по тестам

Существует система Disqus, которая позволяет встраивать комментирование на разрабатываемый ресурс без каких-либо сложностей, система очень проста в использовании, а также очень удобна (рисунок 2). Имеет простой и красивый интерфейс, на главной странице имеется вся справочная информация о данном продукте [2].

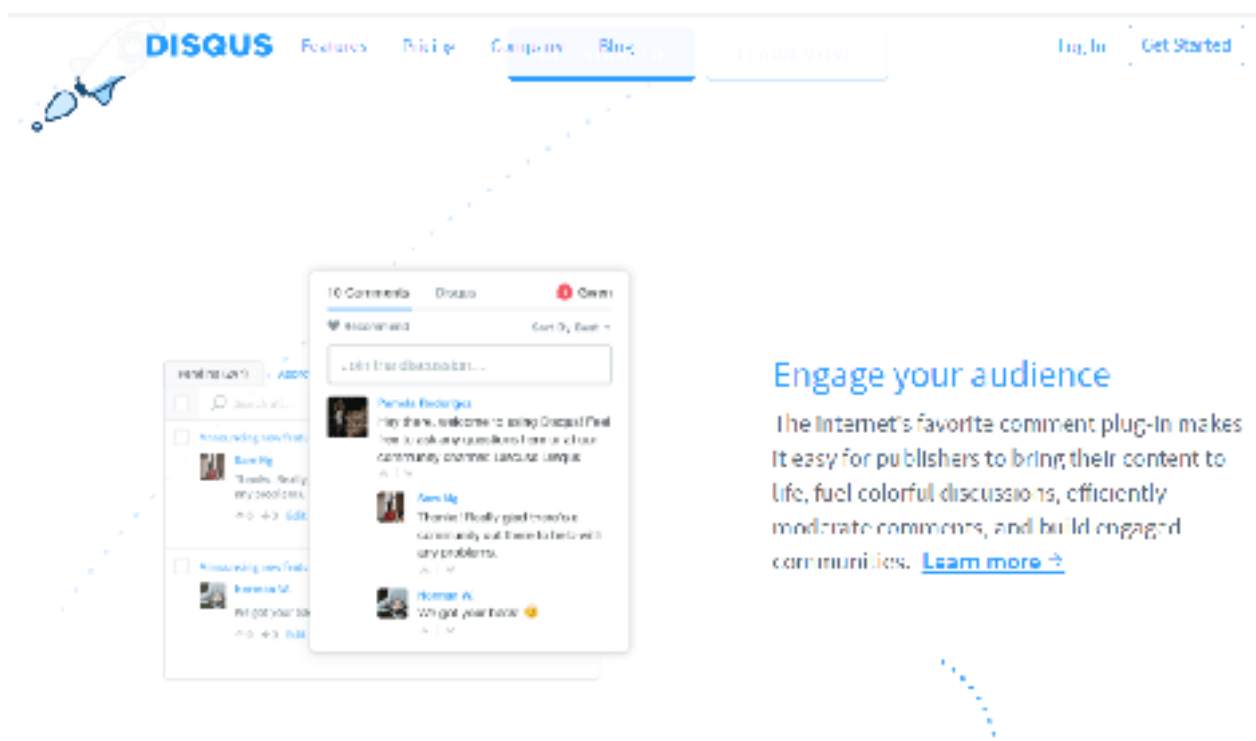


Рисунок 2 – Главная страница системы Disqus с общей информацией о ней

Для создания тестов или опросов существует система Google Forms (рисунок 3), эта система предоставляет большой выбор по функционалу:

- Создание тестов или опросов на любую тему;
- Три типа ответов на вопросы- одиночный, несколько из списка, раскрывающийся;

- Позволяет отслеживать сколько человек прошло созданный вами тест;
- Существует несколько путей распространения теста, посредством ссылки, по электронной почте, и встроить созданный тест.

Данная система общедоступна и поможет решить любую задачу по проведению опроса или тестирования [3].

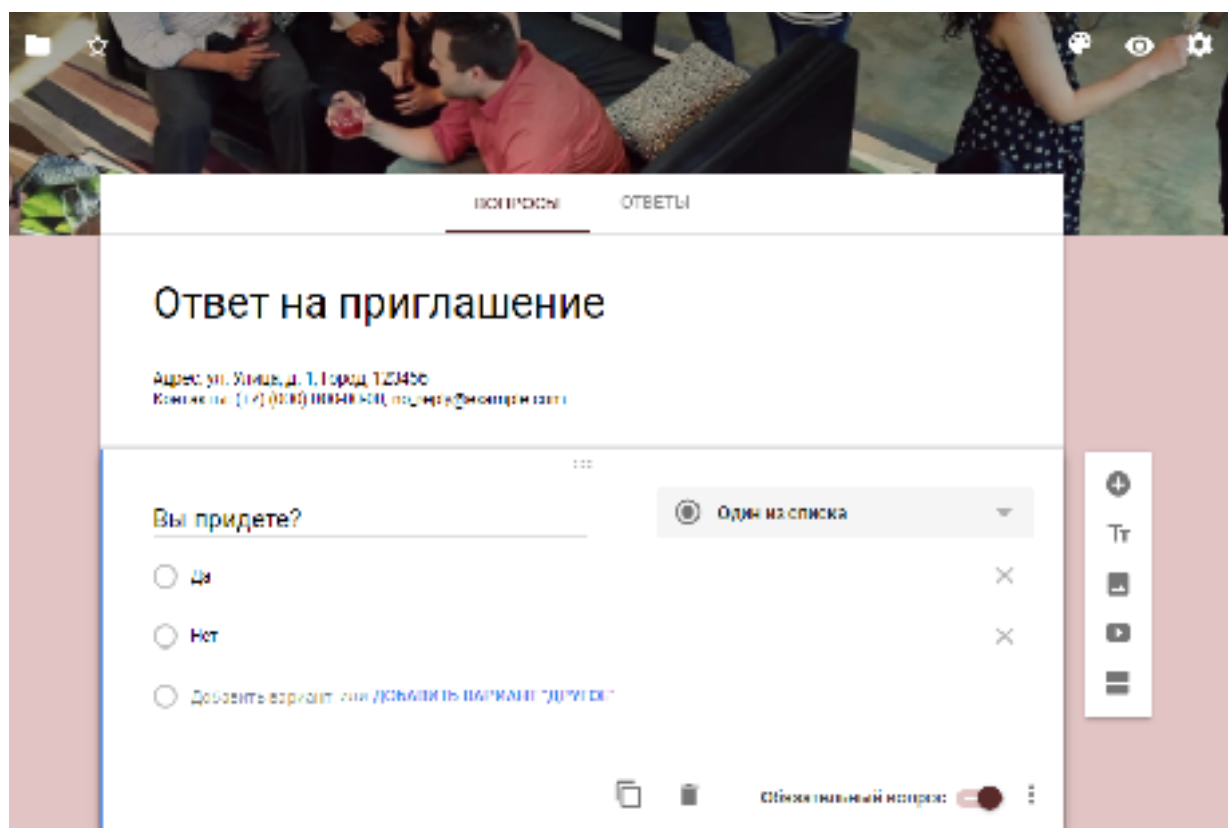


Рисунок 3 – Страница создания опроса в Google Forms

Система «еКурсы». При помощи данной системы возможно создать опрос любого типа, либо тест. Предоставляя доступ ограниченной группе лиц, данная система также ведет статистику и сохраняет результаты пройденных тестов [4]. Система «еКурсы» имеет широкий спектр возможностей по созданию тестов, для данной системы написано несколько модулей.

Однако для того чтобы встроить такую систему необходимо иметь навыки программирования, а также уметь пользоваться сторонним ПО, таким как: Postman, языком программирования PHP, и IDE NetBeans.

Postman - удобный HTTP-клиент для тестирования веб-сайтов (рисунок 4).



Рисунок 4 – Главная страница системы Postman с общей информацией о ней

При помощи данного инструмента можно составлять, редактировать HTTP-запросы любой сложности. Составленные вами запросы автоматически сохраняются для последующего повторного использования. Ответы от сервера также возможно сохранять как файлы. Таким образом, Postman достаточно эффективно экономит время веб-разработчикам и пентестерам.

В программе есть встроенный редактор запросов, с возможностями кодирования запросов, загрузки из файла и отправки бинарных данных.

PHP - это инструмент для создания веб страниц. В более полном определении - это скриптовый язык, который широко применяется для написания веб-приложений (рисунок 5).

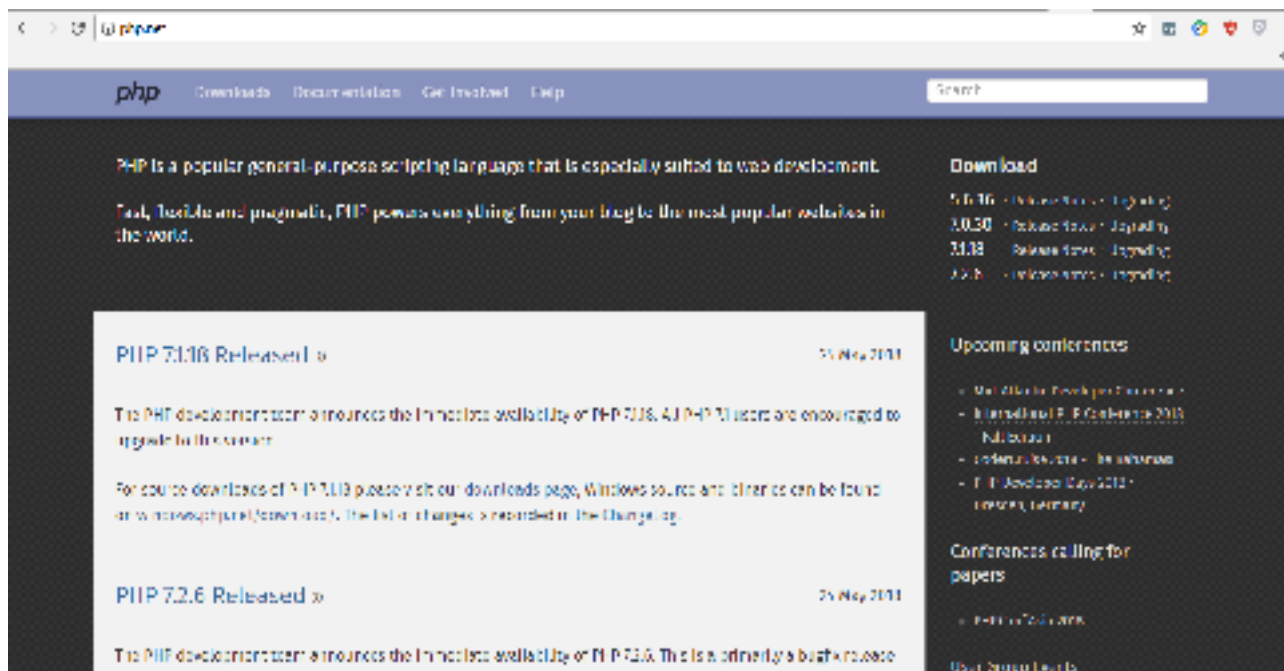


Рисунок 5 – Основной ресурс для изучения языка программирования PHP

На данный момент этот язык поддерживается большинством хостингов и является одним из лидеров языков программирования для создания динамических веб-сайтов.

NetBeans – это (IDE) среда разработки приложений. NetBeans поддерживается Oracle, однако разрабатывается независимым обществом энтузиастов. Данная IDE хорошо подходит для full-stack разработки, то есть для разработки внешнего вида веб-сервиса и back-end (рисунок 6).

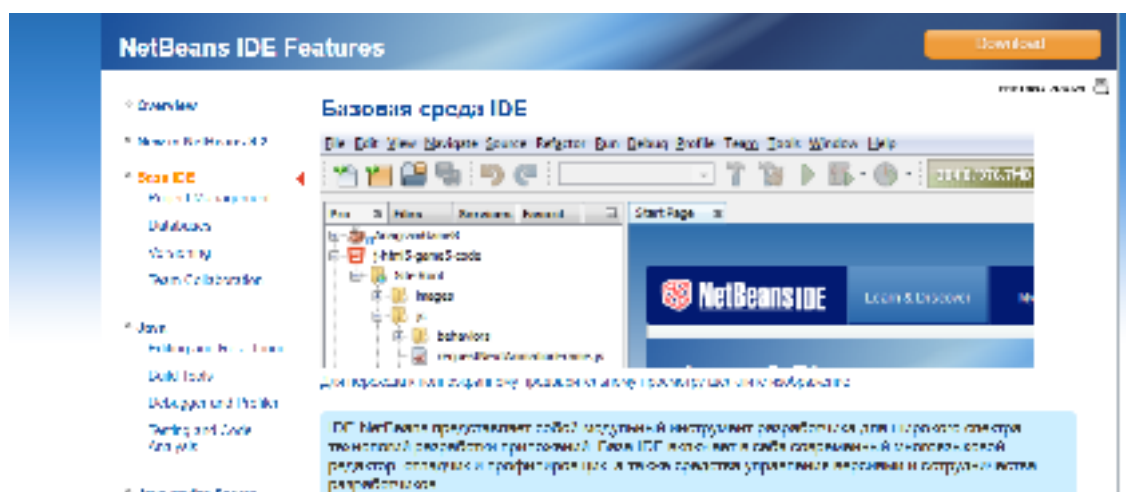


Рисунок 6 – Страница системы NetBeans с общей информацией о ней

Test Form Builder – компонент позволяет создать неограниченное количество тестов с разделением по категориям (рисунок 7).



Рисунок 7 – Страница системы Test Form Builder с общей информацией о ней

Доступ к тестам открыт только зарегистрированным и авторизованным на сайте пользователям.

Каждый тест может содержать неограниченное количество групп вопросов с определением условий дальнейшего прохождения теста.

Выбор ответов на вопросы может быть трёх типов:

- Радиокнопки;
- Чек-боксы;
- Выпадающий список.

В компоненте предусмотрен сбор дополнительных данных пользователей с определением условий доступа к каждому тесту по данным заполненных анкет.

Также имеется ограничение доступа к тесту и задержка времени на повторное прохождение. Предусмотрена настройка доступа к тесту по количеству имеемых положительных и отрицательных результатов.

Условия прохождения теста позволяют определить проходной балл, оценку результата и возможность возврата к предыдущим вопросам и их группам.

Функция ограничения времени может остановить тест при превышении установленного лимита или учесть время прохождения при расчёте результата теста.

Каждый тест имеет настройку публикации по дате и менеджера для просмотра и анализа результатов [5].

Проведя анализ существующих программных средств, учитывая недостатки и достоинства возникает необходимость в создании более удобного сервиса, поэтому будет создана система, которая позволит пользователю встраивать тесты в свой разрабатываемый ресурс.

Разрабатываемая система будет сочетать в себе некоторый функционал вышеприведенных сервисов, создавать тестирование или опрос, и предоставлять его запросившему его ресурсу.

1.2 Разработка системы

В данной работе будут разработаны следующие компоненты:

- Привлекательный простой и несложный веб-интерфейс, понятный любому пользователю, который будет способен собирать, компоновать и отправлять данные о пройденном тестировании на сторону сервера. Данный веб-интерфейс необходимо реализовать так, чтобы пользователь легко мог найти то, что ему требуется.
- Сервер, который будет обрабатывать поступающие запросы, работать с приходящими на него данными, сохранять результаты тестирования. Это будет сделано для ведения статистики, которой в дальнейшем смогут воспользоваться другие сервисы или системы, отдавать на сторону клиента данные о выбранном им тесте.

Кроме этого, необходимо продумать структуру хранения и работы с тестами, формат хранения тестов, добавление новых тестов пользователем.

2 Проектирование

На основе приведенных выше аналогов, исключая их недостатки, будет разработан веб сервис, который будет иметь возможность создания тестов или опросов, а впоследствии их встраивание, кроме того, будет реализована возможность ведения статистики: сколько человек прошло тест, популярность созданный пользователем тестов.

Данная система представляет совокупность двух главных блоков: веб-интерфейс, сервера.

2.1 Модули системы

Данная система состоит из двух блоков: блока сервер, блок интерфейса, каждый из которых будет реализовать поставленные задачи.

При входе и авторизации пользователя на сайт системы, он может ознакомиться со справочной информацией, ознакомиться с уже созданными тестами, создать свой тест, который в последствии возможно будет встроить в другой ресурс.

Посетив данный ресурс и создав тест, пользователь может встроить его в свой разрабатываемый сервис, найти при помощи строки поиска необходимый тест и работать с ним (рисунок 8). Пользователь способен предоставить ссылку на тест, тогда все, у кого будет эта ссылка, смогут пройти предоставленный тест.



Рисунок 8 – Структурная схема работы веб-сервиса

2.1.1 Интерфейс

В блоке веб-интерфейса будет реализован привлекательный внешний вид, система навигации, система авторизации, страница со всеми тестами и опросами, которые можно будет пройти на данном ресурсе, и поиском по ним. Кроме этого, на главной странице любой пользователь сможет получить интересующую его информацию о данном продукте, и ознакомиться с его возможностями.

После прохождения тестирования или опроса, на стороне интерфейса происходит обработка результатов, и демонстрация описания результата пользователю.

Если пользователь хочет создать тест или опрос для своего ресурса, ему необходимо нажать на кнопку «Создать тест» в меню сервиса, создать тест или опрос с интересующим его типом вопросов:

- множественный – пользователь может выбрать несколько ответов;
- данный тип больше подходит для опроса;
- одинарный – существует только один правильный ответ;
- точный – необходимо вписать в поле правильный ответ.

После того, как тест создан, необходимо описать критерии, по которым будет оцениваться результат.

И когда создание теста или опроса полностью завершено, интерфейс отправляет его на сервер, скомпоновав в надлежащий вид.

При помощи языка программирования JavaScript и библиотеки JQuery, в разрабатываемой системе будут созданы обработчики событий нажатия на кнопки, запросы к серверу и обработчики результатов от него, обработка данных пришедших от сервера, работа с DOM элементами, а именно:

- Динамическое создание компонентов - вопросов и ответов в тесте, описания результатов теста;
- Добавление текста в компоненты;
- Создание элементов DOM.

У каждого пользователя будет список тестов, созданных им, помимо этого будет вестись статистика по этим тестам:

- Сколько человек прошли тот или иной тест;
- Популярность каждого теста.

Помимо всего этого пользователю при создании теста будет предоставлена ссылка для раздачи ее кругу лиц. Это создано для того, если разработчик не хочет или не может встроить тест, он может предоставить данную ссылку группе лиц, и они смогут пройти данный тест в данной системе.

2.1.2 Сервер

Сервер - это в некотором роде мозг данной системы, он слушает поступающие на него запросы, обращается к базе за тестами или сохраняет их в ней. А также при авторизации пользователей сервер проверяет существование логина и пароля авторизующегося пользователя. Работает со статистикой, также посредством базы данных.

После того, как на сервер поступили данные созданного пользователем теста, сервер создает персональный идентификатор полученного теста или опроса и сохраняет его в базу данных, а после, отправляет на интерфейс, сформированный URL с идентификатором для встраивания данного теста или опроса на стороннем ресурсе. Также сервер способен «слушать» обращения к нему и по определенному URL отдавать html страницы. Если на сервер пришел запрос о тесте, то он, обратившись к базе и получив запрашиваемые данные по ключу, который находится в адресе запроса, отправляет модуль с тестом или опросом на ресурс, который эти данные запросил.

Блок сервера будут реализовывать некую систему работы с данными: хранение, обработка, принятие, отправка. Все это необходимо для ведения статистики, а также для встраивания тестов и опросов в сторонние ресурсы.

Помимо всего этого сервер способен, по определенному запросу, перенаправлять пользователей на прохождение теста на разрабатываемом ресурсе, такое поведение возможно только в том случае, если создатель теста выбрал пункт «Тестирование на данном ресурсе».

3 Реализация

3.1 Используемые инструменты

Разрабатываемая система была написана при помощи редактора кода VisualStudioCode – редактор исходного кода, разработанного Microsoft для Windows, Linux, macOS. Примененный редактор позиционируется как «лёгкий» для кроссплатформенной разработки веб и облачных приложений. Данный редактор включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации [6].

3.1.1 Инструменты интерфейса

Разработанная система была реализована посредством языка программирования JavaScript.

Язык программирования JavaScript поддерживает объектно-ориентированный, императивный и функциональный стиль [7]. Архитектурными чертами использованного языка программирования являются: динамическая типизация, автоматическое управление памятью, прототипное программирование, функции как объекты первого класса. С его помощью доступны к исполнению следующие функции:

- возможность изменять страницы браузеров;
- добавление или удаление тегов;
- изменение стилей страницы;

- наличие информации о действиях пользователя на странице;
- запрос доступа к случайной части исходного кода страницы;
- внесение изменений в этот код.

Область применения этого языка удивительно обширна и ничем не ограничена: среди программ, которые используют JS, присутствуют и текстовые редакторы, и приложения (как для компьютеров, так и мобильные и даже серверные), и прикладное ПО.

В разработке системы были использованы библиотеки JQuery и набор инструментов Bootstrap.

JQuery — это библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими, и языком программирования JavaScript [8].

Bootstrap (известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

При помощи библиотеки JQuery были созданы обработчики событий нажатий на кнопки меню и другие элементы.

Средствами CSS была достигнута адаптивность сервиса под разные размеры экрана и браузера. При помощи фреймворка Bootstrap было создано меню сайта, при помощи данного инструмента была достигнута адаптивность компонентов страницы. Благодаря использованным инструментам у пользователя появляется возможность комфортно работать с системой.

3.1.2 Инструменты сервера

Сервер реализован при помощи NodeJS – это программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения [9]. Помимо NodeJS использовался шаблонизатор EJS. Шаблонизатор- это программное обеспечение, позволяющее использовать html-шаблоны для генерации конечных html-страниц [10]. Основная цель использования шаблонизаторов — это отделение представления данных от исполняемого кода. Часто это необходимо для обеспечения возможности параллельной работы программиста и дизайнера-верстальщика. Использование шаблонизаторов улучшает читаемость кода и внесение изменений во внешний вид.

В данной системе использовался минималистический веб-фреймворк Express.js, или просто Express, каркас web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT [11]. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js. Автор фреймворка, TJ Holowaychuk, описывает его как созданный на основе написанного на языке Ruby каркаса Sinatra, подразумевая, что он минималистичен и включает большое число подключаемых плагинов. Express может являться backend'ом для программного стека MEAN, вместе с базой данных MongoDB и каркасом AngularJS для frontend'a.

3.2 Описание работы системы

3.2.1 Интерфейс

На главной странице пользователя будет ожидать некое описание и панель навигации, которая находится на верхней строке страницы, здесь же находится и логотип (рисунок 9).



Рисунок 9 – Функциональная схема работы интерфейса.

Кнопка «Домой» – позволит пользователю вернуться на главную страницу сервиса (рисунок 10).

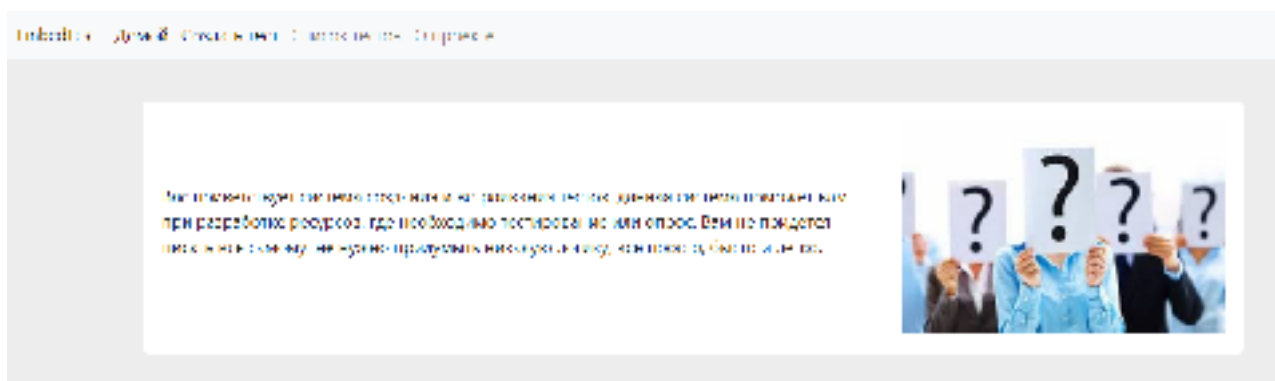


Рисунок 10 – Главная страница сервиса

Кнопка «Тесты» – дает возможность просмотра уже существующих тестов (рисунок 11).



Рисунок 11 – Список тестов

Кнопка «О проекте» – на данной странице пользователь сможет ознакомиться с данным ресурсом, его функциональностью и способом пользования, а также узнать о разработчиках данной системы.

Кнопка «Создание теста» – позволит создать собственный тест, которого нет на этом сервисе, только авторизованным пользователям (рисунок 12).

The screenshot shows the 'Создание теста' (Create Test) page of the EmbedTest application. The top navigation bar includes links for 'EmbedTest', 'Домой', 'Создание теста', 'Список тестов', and 'О проекте'. On the left side, there are three buttons: 'Добавить вопрос', 'Сохранить тест', and 'Удалить вопрос'. The main content area is titled 'Название теста' and contains a large input field labeled 'Введите вопрос' with a '+' and '-' icon to its right. Below this, there are two columns: 'Ответ' and 'Цена'. The 'Ответ' column has a sub-label 'Ссылка' and a text input field. The 'Цена' column has a sub-label 'Цена' and a text input field. At the bottom, there are two buttons: 'Добавить критерий' and 'Удалить критерий'. Below these buttons is a table with two columns: 'Критерий' and 'Ответ'. The 'Критерий' column has two rows with text input fields, and the 'Ответ' column has a single text input field.

Рисунок 12 – Создание теста

Для создания теста необходимо будет ввести некоторые данные:

- Название теста (в заголовке теста, также в поле ввода, предназначенном для этого, имеется подсказка);
- Создать требуемое количество вопросов при помощи кнопки с соответствующим названием, и заполнить созданные поля.
- В каждом вопросе теста создать необходимое количество ответов, это осуществляется при помощи нажатия на кнопку рядом с вопросом, и ввести ответы и балл за каждый ответ (цена ответа).

- После внесения всех данных сохранить тест, после чего пользователю будет предоставлена ссылка на данный тест.

Кроме этого, если пользователю понравился другой тест, то он может открыть данный тест и получить его уникальную ссылку (рисунок 13).

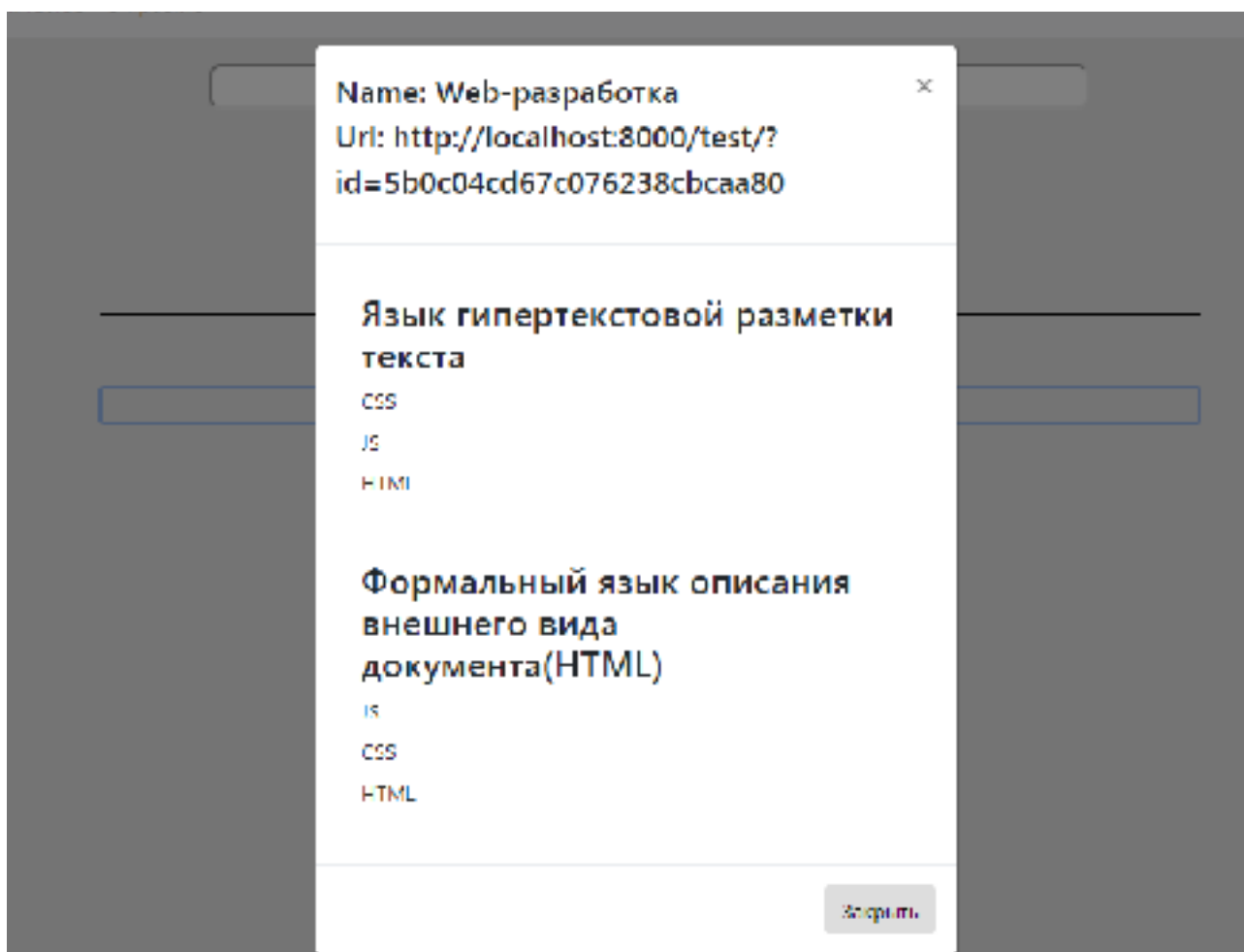


Рисунок 13 – Модальное окно с выбранным тестом

3.2.2 Сервер

В этом блоке осуществляется принятие и сохранение данных (рисунок 14).



Рисунок 14 – Структурная схема работы сервера

Сервер умеет «слушать» поступающие на него запросы и обрабатывать их. На сервере реализованы следующие обработчики «событий»:

- Обработчик на отправку домашней страницы клиенту.
- Обработчик на добавление теста.
- Обработка запроса на получение конкретного теста по его уникальному ID.

- Обработчик для показа содержимого теста по клику на него в списке тестов.
- Отображение страницы, где будет располагаться список всех тестов.
- Запрос списка всех тестов.
- Поиск по названию теста.

Также сервер совершает запросы в базу данных MongoDB – это документоориентированная система управления базами данных (СУБД) с открытым исходным кодом, не требующая описания схемы таблиц. Классифицирована как NoSQL, использует JSON-подобные документы и схему базы данных. Написана на языке C++ [12].

При разработке встраиваемого модуля, был использован шаблонизатор EJS. С его помощью была создана HTML страница, в которую отправлялись данные с сервера, страница заполнялась ими. На этой же странице происходит обработка пройденного теста и выводится результат.

При создании тестов на сервер приходит объект, который в дальнейшем будет отправлен в базу. При поиске теста, при выводе списка тестов или при встраивании теста, также происходит запрос в базу с определенными атрибутами поиска. После чего на интерфейс, в зависимости от запроса, приходит или массив тестов, или объект – запрошенный тест.

3.3 Описание базы данных

База данных реализована посредством MongoDB. Она осуществляет хранилище всех зарегистрированных пользователей и созданных ими тестов и опросов.

Пользователи хранятся в виде массивов объектов с некоторыми полями:

- Уникальный идентификатор;

- Логин;
- Пароль;
- Созданные тесты;
- Созданные опросы.

Все тесты хранятся как один массив объектов (рисунок 15). Каждый объект хранит в себе поля, которые пользователь заполнял при создании теста, опроса:

- Название теста;
- Поле вопросов – массив объектов, в каждом из которых находится текст вопроса и текст ответов на него, а также id данного вопроса;
- Поле цены- оценка каждого ответа на вопрос;
- Поле описания- описание результатов теста.

У каждого теста будет вестись некоторая статистика, такая как:

- Популярность данного теста;
- Количество пользователей, которые прошли данный тест.

```

{
  "_id": {
    "$oid": "5b0c04cd67c076238cbcaa80"
  },
  "nameTest": "Web-разработка",
  "question": [
    {
      "text": "Язык гипертекстовой разметки текста",
      "id": "0",
      "answer": [
        "CSS",
        "JS",
        "HTML"
      ]
    },
    {
      "text": "Формальный язык описания внешнего вида документа(HTML)",
      "id": "1",
      "answer": [
        "JS",
        "CSS",
        "HTML"
      ]
    }
  ],
  "price": [
    [
      "0",
      "0",
      "1"
    ],
    [
      "0",
      "1",
      "0"
    ]
  ],
  "description": [
    {
      "chapel-in": "1",
      "chapel-out": "2",
      "description": "Ты знаешь основы Web-разработки"
    },
    {
      "chapel-in": "0",
      "chapel-out": "0",
      "description": "Ты не знаешь основы Web-разработки"
    }
  ]
}

```

Рисунок 15 – Объект теста в базе данных

4 Результат работы

В данной работе был разработан сервис по созданию и встраиванию тестов (рисунок 16).



Рисунок 16 – Структурная схема работы встраиваемого компонента тестирования

Любой пользователь может посетить данную систему, создать необходимый ему опрос или тест и встроить его в свой веб-сайт или другой интернет ресурс, кроме того присутствует возможность предоставить уникальную ссылку другим пользователям для прохождения тестирования или опроса в данной системе.

Для того, чтобы встроить тест в свой ресурс пользователю необходимо посетить разрабатываемый веб сервис, создать необходимый тест или опрос, получить уникальную ссылку, и использовать ее в своем ресурсе.

При создании собственного ресурса пользователь должен сделать запрос по уникальному URL, полученному при создании теста, и вставить результат запроса в созданный для этого тег (рисунок 17).

Пользователь без особого труда может встроить в свой веб ресурс любой тест или опрос за пару простых действий, разрабатываемая система очень проста в использовании и не требует установки стороннего ПО.

```
1 <script src="https://code.jquery.com/jquery-3.3.1.js"
2 integrity="sha256-2Kok/Nb0yxpjNvAk/H12jig0V52auKAPf7bn/0Hf8=" crossorigin="anonymous">
3 </script>
4 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
5 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
6 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
7 <div class="myTest">
8
9 </div>
10 <script>
11     $.ajax({
12         type: "GET",
13         url: "http://localhost:8080/test/?id=5b9c84cd6/c8/62d8cbca288",
14     }).done(data => [
15         $(".myTest").html(data);
16     ]).fail(error => console.log(error))
17 </script>
```

Рисунок 17 – Код для встраивания теста на стороннем ресурсе

Результатом является отображение теста, первый вопрос отображается сразу (рисунок 18), каждый последующий появляется при ответе на предыдущий (рисунок 19).

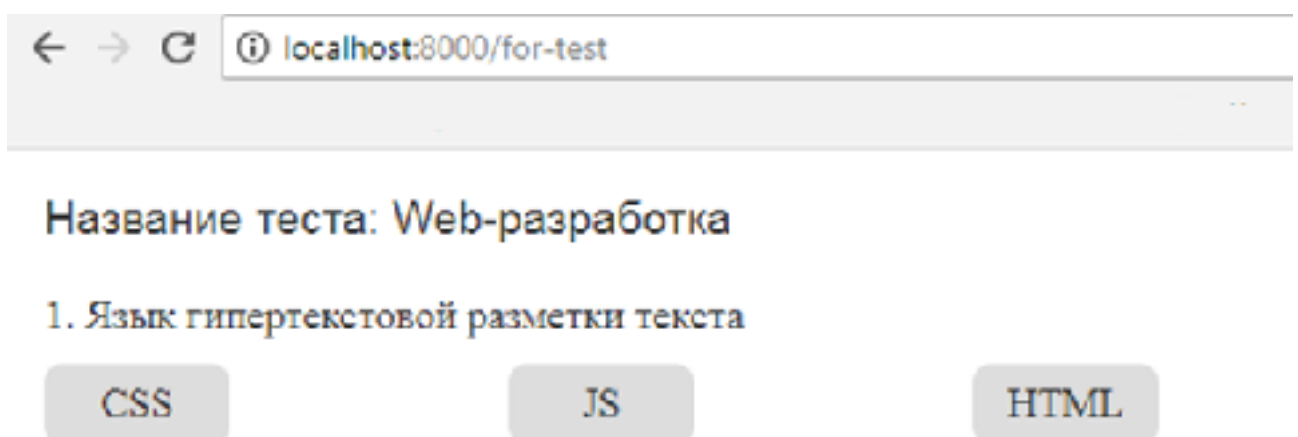


Рисунок 18 – Первый вопрос встроенного теста

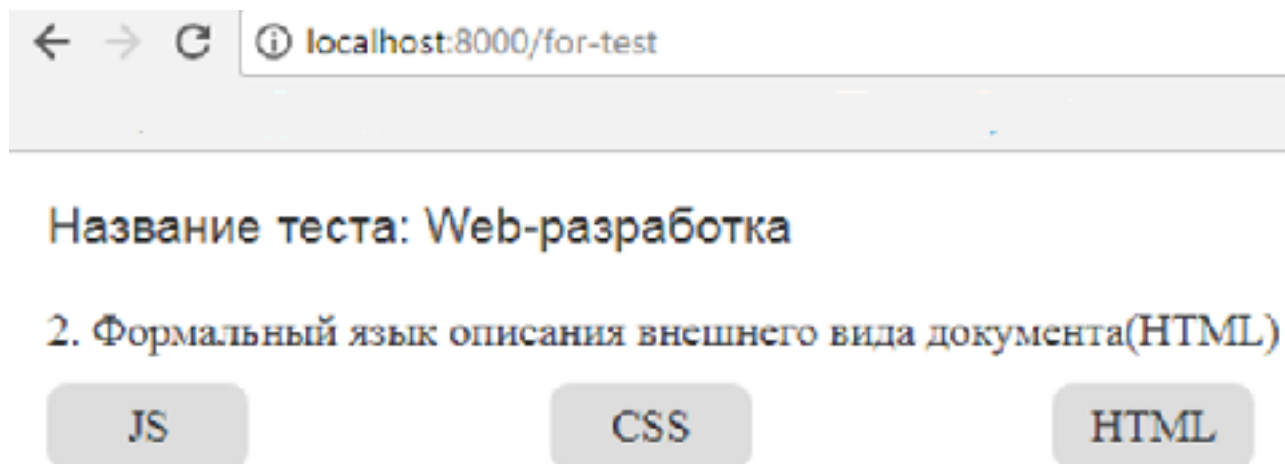


Рисунок 19 – Второй вопрос созданного теста

ЗАКЛЮЧЕНИЕ

Результатом данной работы является создание сервиса коротких тестов, с использованием языка программирования JavaScript и библиотеки JQuery, фреймворка Bootstrap.

Кроме этого, реализован встраиваемый модуль, который обеспечивает проведение опроса или тестирования аудитории на стороннем ресурсе.

Реализована система ведения статистики по созданным тестам, архитектура хранения данных, приятный в использовании интерфейс, сервер работающий с запросами от интерфейса и базой данных.

Данная система призвана решать проблему разработчика по созданию своего собственного сервиса тестирования.

Разработанная система коротких тестов сэкономит разработчику время и ресурсы, позволит создавать тесты любого типа, разработчику требуется лишь совершить несколько простых действий. Кроме того, разработчик сможет использовать уже созданные им тесты либо опросы, которые хранятся в базе данных.

Следует отметить преимущество системы – система проста в использовании и имеет несложный интерфейс.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Сервис создания тестов [Электронный ресурс] : – Режим доступа: <https://www.surveymonkey.ru/>
2. Сервис встраивания комментирования [Электронный ресурс] : – Режим доступа: <https://disqus.com/>
3. Google Forms [Электронный ресурс] : – Режим доступа: <https://docs.google.com/forms>
4. Система электронного обучения СФУ [Электронный ресурс] : – Режим доступа: <https://e.sfu-kras.ru/>
5. Система создания тестов J R E x t e n s i o n s [Электронный ресурс] : – Режим доступа: <http://www.jrextensions.ru/test-form-builder/test-form-builder-3>
6. Редактор кода VisualStudioCode [Электронный ресурс] : – Режим доступа: <https://code.visualstudio.com/>
7. Современный учебник JavaScript [Электронный ресурс] : – Режим доступа: <https://learn.javascript.ru/>
8. Библиотека JavaScript [Электронный ресурс] : – Режим доступа: <http://jquery.com/>
9. Сайт программной платформы NodeJs [Электронный ресурс] : – Режим доступа: <https://nodejs.org/en/>
10. Википедия Свободная энциклопедия [Электронный ресурс] : – Режим доступа: <https://ru.wikipedia.org/wiki/Шаблонизатор>
11. Википедия Свободная энциклопедия [Электронный ресурс] : – Режим доступа: <https://ru.wikipedia.org/wiki/Express.js>
12. Википедия Свободная энциклопедия [Электронный ресурс] : – Режим доступа: <https://ru.wikipedia.org/wiki/MongoDB>